

7/5/4 (Item 2 from file: 347)
DIALOG(R) File 347:JAPIO
(c) 2004 JPO & JAPIO. All rts. reserv.

03371026 **Image available**
MICROCOMPUTER

PUB. NO.: 03-033926 [JP 3033926 A]
PUBLISHED: February 14, 1991 (19910214)
INVENTOR(s): MIYAZAWA AZUMA
ISHIMARU HISAAKI
APPLICANT(s): OLYMPUS OPTICAL CO LTD [000037] (A Japanese Company or
Corporation), JP (Japan)
APPL. NO.: 01-131861 [JP 89131861]
FILED: May 25, 1989 (19890525)
INTL CLASS: [5] G06F-009/06; G06F-015/78
JAPIO CLASS: 45.1 (INFORMATION PROCESSING -- Arithmetic Sequence Units);
45.4 (INFORMATION PROCESSING -- Computer Applications)
JAPIO KEYWORD: R131 (INFORMATION PROCESSING -- Microcomputers &
Microprocessors)
JOURNAL: Section: P, Section No. 1196, Vol. 15, No. 169, Pg. 32, April
26, 1991 (19910426)

ABSTRACT

PURPOSE: To rewrite part of a ROM falsely by storing address data desiring to correct in the ROM into a nonvolatile memory, and executing a modified program when the data coincides with a program counter.

CONSTITUTION: The address data and correction program data when a bug or partial correction is desired for the writing program of a program memory ROM 2 are written on an electrically writable nonvolatile memory EPROM 3. When the address data coincides with the output content of the program counter 1, corrected program data is outputted from the EPROM 3 instead of the program data to be corrected in the memory 2. And the corrected program data is selected with a selector 4, and is inputted to an instruction decoder 5. Therefore, the program rewritten falsely is decoded with the instruction decoder 5, and a corrected program can be executed.

⑫ 公開特許公報(A) 平3-33926

⑤ Int.Cl.³G 06 F 9/06
15/78

識別記号

440 N
510 A

庁内整理番号

7361-5B
9072-5B

⑬ 公開 平成3年(1991)2月14日

審査請求 未請求 請求項の数 1 (全11頁)

⑭ 発明の名称 マイクロコンピュータ

⑰ 特 願 平1-131861

⑱ 出 願 平1(1989)5月25日

優先権主張 ⑳ 平1(1989)3月28日㉑ 日本(JP)㉒ 特願 平1-76047

⑲ 発 明 者 宮 沢 東 東京都渋谷区幡ヶ谷2丁目43番2号 オリンパス光学工業株式会社内

⑲ 発 明 者 石 丸 寿 明 東京都渋谷区幡ヶ谷2丁目43番2号 オリンパス光学工業株式会社内

⑳ 出 願 人 オリンパス光学工業株式会社 東京都渋谷区幡ヶ谷2丁目43番2号

㉑ 代 理 人 弁理士 坪 井 淳 外2名

明 細 書

1. 発明の名称

マイクロコンピュータ

2. 特許請求の範囲

プログラムカウンタ、リードオンリメモリ、命令デコードと電気的に書き込み可能な不揮発性メモリを有するマイクロコンピュータにおいて、

前記不揮発性メモリに前記リードオンリメモリの修正したいアドレスに対応するアドレスデータを記憶させ、前記プログラムカウンタと前記不揮発性メモリに記憶されたアドレスデータとが一致したときは変更したプログラムを実行することを特徴とするマイクロコンピュータ。

3. 発明の詳細な説明

〔産業上の利用分野〕

本発明は、マイクロコンピュータに係り、詳しくはリードオンリメモリの修正したいアドレスの実行時に変更したプログラムを実行するマイクロコンピュータに関する。

〔従来の技術〕

一般的に、従来の1チップマイクロコンピュータは、第12図に示すように、プログラムカウンタPC、所定のプログラムが書込まれたプログラムメモリPMおよび命令デコーダIDを有する。この1チップマイクロコンピュータの動作はよく知られているように、システムクロック(図示せず)によってカウントアップするプログラムカウンタPCによってアドレスが指定されるプログラムメモリPMから読出されるデータが命令デコーダIDによってデコードされることによってプログラムが実行される。

プログラムメモリPMは、一般的にマスクROM(リードオンリーメモリ)が用いられ、このマスクROMはメーカーの製造工程でメーカーによりプログラムが書込まれ、ユーザによる書換えは不可能である。

また、最近では、ユーザによる書換えが1回だけ可能なワンタイムROMと呼ばれるものも販売されている。

〔発明が解決しようとする課題〕

しかし、1チップマイクロコンピュータの製造後にマスクROMの書き込みプログラムにバグなどが発見された場合やその一部を修正したい場合、再度、ROMのマスクパターンを書換えてから集積回路の製造プロセスをやり直す必要があるため、修正された1チップマイクロコンピュータを入手するまでの期間が現状では数ヶ月もかかる。

しかも、マスクROMの書き込みプログラムにバグなどが発見された1チップマイクロコンピュータは再利用することができない。また、ワンタイムROMはマスクROMに比べて非常に高価であり、ユーザがプログラムを書込むのに非常に時間がかかり、量産品への採用には不向きである。

本発明は、このような課題に着目してなされたもので、リードオンリメモリの書き込みプログラムにバグなどが発見された場合やその一部を修正したい場合に、リードオンリメモリの内容を書換えることなく、一部を疑似的に書換えてプログラムを修正でき、あるいは、割込み処理を行わせるこ

スデータとプログラムカウンタとが一致した時は、リードオンリメモリの修正すべきプログラムデータに代えて不揮発性メモリに記憶されている修正されたプログラムデータが命令デコードに出力されるようになり、バグなどが修正されたプログラムが実行される。

また、リードオンリメモリの書き込みプログラムにバグなどが発見された場合やその一部を修正したい場合に、このリードオンリメモリの修正したいアドレスに対応するアドレスデータと修正するための割込み用のプログラムデータを不揮発性メモリに記憶させておくことにより、不揮発性メモリに記憶されたアドレスデータとプログラムカウンタとが一致した時は、リードオンリメモリの修正すべきプログラムデータに代えて不揮発性メモリに記憶されている割込み用のプログラムデータに基づいて割込み処理が行われる。この割込み処理では、メモリのデータの書換えとか、追加したい機能のサブルーチンのコールとか、無条件ジャンプなどによりそれまでの処理の流れを変え、実

により実質的にプログラムの追加とか削除を行ない得るマイクロコンピュータを提供することを目的とする。

〔課題を解決するための手段〕

本発明のマイクロコンピュータは、プログラムカウンタ、リードオンリメモリ、命令デコードと電氣的に書き込み可能な不揮発性メモリを有するマイクロコンピュータにおいて、上記不揮発性メモリに上記リードオンリメモリの修正したいアドレスに対応するアドレスデータを記憶させ、上記プログラムカウンタと上記不揮発性メモリに記憶されたアドレスデータとが一致したときは変更したプログラムを実行するようにしている。

〔作用〕

リードオンリメモリの書き込みプログラムにバグなどが発見された場合やその一部を修正したい場合に、このリードオンリメモリの修正したいアドレスに対応するアドレスデータと修正されたプログラムデータを不揮発性メモリに記憶させておくことにより、不揮発性メモリに記憶されたアドレ

質的にプログラムの追加とか削除を行なうことが可能であり、プログラムの修正が可能になる。

〔実施例〕

以下、図面を参照して本発明の実施例を詳細に説明する。

まず、本発明のマイクロコンピュータの概念を第1図を参照して説明する。第1図は、たとえば1チップのマイクロコンピュータを示しており、1はプログラムカウンタ、2は例えばマスクROMが用いられた所定のプログラムが書き込まれたプログラムメモリ、3は電氣的に書き込み可能な不揮発性メモリ（たとえばEPROM、EEPROMなどであり、以下、EROMと略記する）、4はセレクト、5は命令デコードである。

プログラムカウンタ1は、プログラムメモリ2だけでなく、EROM3にもアドレス値を与えるように接続されている。EROM3は、プログラムメモリ2の書き込みプログラムにバグなどが発見された場合やその一部を修正したい場合に、このプログラムメモリ2の修正したいアドレスに対応

するアドレスデータと修正するためのプログラムデータが書込まれている。セレクト4は、通常はプログラムメモリ2の出力データを選択して命令デコーダ5に入力させるが、EROM3に記憶されたアドレスデータとプログラムカウンタ1の出力内容とが一致したときにEROM3からプログラムデータが出力すると、このEROM3の出力データ（修正されたプログラムデータ）をプログラムメモリ2の出力データ（修正すべきプログラムデータ）に代えて選択して命令デコーダ5に入力させるように構成されている。

次に、第1図のマイクロコンピュータの動作を説明する。このマイクロコンピュータの動作は、基本的には従来のマイクロコンピュータの動作と同様であるが、さらに、プログラムメモリ2の書き込みプログラムの一部が疑似的に書き換えられてプログラムが修正されるようになっている。

すなわち、通常は、システムクロック（図示せず）によってカウントアップするプログラムカウンタ1によってアドレスが指定されるプログラム

により構成されており、その他は第1図に示したマイクロコンピュータと同じであり、第1図中と同一部分には同一符号を付してその説明を省略する。

第2図の1チップマイクロコンピュータの動作は、第1図を参照して前述した動作と同様であるのでその詳述は省略し、以下、主として、書き込み制御部6の動作について説明する。プログラムメモリ2の書き込みプログラムにバグなどが発見された場合やその一部を修正したい場合に、書き込み制御部6からEROM3に書き込み許可信号を与えると、EROM3はデータ書き込みが可能になる。さらに、書き込み制御部6からEROM3に対して、EROMアドレスを指定するとともにEROMデータを与え、書き込み信号を活性化すると、EROM3には上記EROMデータが書込まれる。この場合、EROMデータは、前述したようなアドレスデータおよび修正用のプログラムデータであり、このアドレスデータおよび修正用のプログラムデータは同じEROMアドレスに同時に書込

メモリ2から読出されるデータがセレクト4により選択されて命令デコーダ5によってデコードされることによってプログラムが実行される。しかし、EROM3に記憶されているアドレスデータとプログラムカウンタ1の出力内容とが一致したときにEROM3からプログラムデータが出力すると、このEROM3の出力データ（修正されたプログラムデータ）がプログラムメモリ2の出力データ（修正すべきプログラムデータ）に代わってセレクト4により選択されて命令デコーダ5に入力する。これにより、疑似的に書き換えられたプログラムが命令デコーダ5によってデコードされ、修正されたプログラムが実行されることになる。

第2図および第3図は、本発明のマイクロコンピュータの第1実施例を示している。第2図は、たとえば1チップのマイクロコンピュータを示しており、6はEROM3に対する書き込みを行なうための書き込み制御部であり、セレクト4'はEROM3からデータ切換制御信号を受けてデータ選択を行なうように、たとえばマルチプレクサ

まれてもよく、別々のEROMアドレスに順次に書込まれてもよい。

第3図は、第2図中のEROM3の一部の具体例を示しており、所望のデータ修正数に対応して設けられた複数（ n ）個のデータ修正ブロック31～3 n を有する。各ブロック31～3 n において、それぞれ電氣的に書き込み可能な不揮発性メモリ素子が用いられた一対のメモリ領域としてアドレスデータ領域32およびプログラムデータ領域33が確保されており、このアドレスデータ領域32およびプログラムデータ領域33は、プログラムメモリ2の修正すべきプログラムデータが書込まれている特定のアドレスに対応するアドレスデータおよび、この特定のアドレスの内容として修正されたプログラムデータがそれぞれ対応して書込まれる。

さらに、アドレスデータ領域32およびプログラムデータ領域33に対して1つのデータ判定回路34が設けられており、このデータ判定回路34は、プログラムカウンタ1からのアドレス入

力とアドレスデータ領域32に記憶されているアドレスとを比較判定し、一致時には一致信号出力を活性化してプログラムデータ領域33に記憶されているプログラムデータ（修正データ）を出力させるように制御する。

なお、各ブロック31～3nが受け持つデータ修正箇所（アドレス）は互いに異なるので、データ修正時にはどれか1つのブロックから修正データが出力される。

そして、各ブロック31～3nの出力データ（修正データ）は、共通のバスを経てセレクト4のデータ入力となり、各ブロック31～3nの一致信号出力はオア回路35に入力し、このオア回路35の出力はセレクト4に対して切換制御入力となり、このEROM3の出力データがプログラムメモリ2の出力データ（修正すべきプログラムデータ）に代えて選択させる。

なお、EROM3は、プログラムメモリ2の全てのアドレスをカバーできるので、EROMの容量は少なくても済み、EROMを付加したことによ

マルチプレクサ42に入力する。データ判定回路41は、EROM3'のアドレス不一致時に出力する固定データと同じ固定データを格納しており、この格納データと入力データとを比較判定し、不一致時（データ修正の必要がある時）には切換制御信号出力を非活性状態にし、一致時（データ修正の必要がない時）には切換制御信号出力を活性化する。

マルチプレクサ42は、データ判定回路41からの切換制御信号入力が入力が非活性状態の時（データ修正の必要がある時）には、EROM3'からの修正データをプログラムメモリ2の出力データ（修正すべきプログラムデータ）に代えて選択して出力し、データ判定回路41からの切換制御信号入力が活性状態の時（データ修正の必要がない時）には、プログラムメモリ2の出力データ（この時はデータ修正の必要がないプログラムデータ）をそのまま選択して出力する。

第5図は、本発明のマイクロコンピュータの第3実施例を示している。第5図は、たとえば1チ

ップマイクロコンピュータの価格の上昇は僅かで済む。

次に、本発明のマイクロコンピュータの第2実施例を説明する。この第2実施例は、第1実施例と比べて、EROM3'およびセレクト4'が異なり、その他は同じであるので第1実施例中と同一符号を付している。EROM3'は、第3図中のオア回路35が省略され、データ切換制御信号を出力しないように構成されており、アドレス比較判定の結果、一致時にはプログラムデータ領域33から修正データを出力するが、不一致時（データ修正の必要がない時）には所定の固定データ（たとえばNOPコードなど）を出力するように構成されている。

セレクト4'は、上記データ切換制御信号を受けることなくデータ選択を行なうように、たとえば第4図に示すように構成されている。すなわち、第4図において、EROM3'からのデータはデータ判定回路41およびマルチプレクサ42に入力し、また、プログラムメモリ2からのデータは

ップのマイクロコンピュータを示しており、第2図を参照して前述したマイクロコンピュータと比べて、(a) EROM53は、プログラムメモリ2の書き込みプログラムにバグなどが発見された場合やその一部を修正したい場合に、このプログラムメモリ2の修正したいアドレスに対応するアドレスデータと修正するための割込み処理用のプログラムデータが書き込まれている点、(b) プログラムカウンタ1の出力（アドレス値）およびEROM53に記憶されているアドレスデータが一致検出部54に与えられている点、(c) 一致検出部54が両入力の一致を検出した時に、EROM53に記憶されている割込み処理用のプログラムデータが割込み発生回路55に入力し、この割込み発生回路55によりプログラムカウンタ1の値の書き換えを行うように制御する点、(d) セレクト（第2図4）が省略され、プログラムメモリ2から読出されるデータがセレクトを介することなく命令デコーダ5に入力している点と異なり、その他は同じであるので第2図中と同一符号

を付している。なお、EROM 32は、本マイクロコンピュータのシステムバス（図示しない）に接続されている。

次に、第5図のマイクロコンピュータの動作を説明する。このマイクロコンピュータの動作は、基本的には従来のマイクロコンピュータの動作と同様であるが、任意のプログラム位置で割込み処理を発生させ、この割込み処理によりプログラムを修正できるようになっている。

すなわち、まず、書き込み制御部56の動作について説明する。プログラムメモリ2の書き込みプログラムにバグなどが発見された場合やその一部を修正したい場合に、書き込み制御部56からEROM 53に書き込み許可信号を与えると、EROM 53はデータ書き込みが可能になる。さらに、書き込み制御部56からEROM 53に対して、EROMアドレスを指定するとともにEROMデータを与え、書き込み信号を活性化すると、EROM 53には上記EROMデータが書き込まれる。この場合、EROMデータは、前述した

ようなアドレスデータおよび修正するための割込み用のプログラムデータであり、このアドレスデータおよび割込み用のプログラムデータは同じEROMアドレスに同時に書き込まれてもよく、別々のEROMアドレスに順次に書き込まれてもよい。

そして、通常の動作時には、システムクロック（図示せず）によってカウントアップするプログラムカウンタ1によってアドレスが指定されるプログラムメモリ2から読出されるデータが命令デコード5によってデコードされることによってプログラムが実行される。しかし、EROM 53に記憶されているアドレスデータとプログラムカウンタ1の出力内容とが一致したときに割込みが発生し、プログラムカウンタ1の値が書き換えられ、割込み処理が行われる。この場合、EROM 53から出力する修正するための割込み用のプログラムデータに基づいて各種の処理（メモリのデータの書き換えとか、追加したい機能のサブルーチンのコールとか、無条件ジャンプなど）が行われることにより、それまでの処理の流れが変えられ、実

質的にプログラムの追加とか削除が行われることになる。

上述したように、前記第1実施例および第2実施例のマイクロコンピュータでは、修正前のプログラムのステップ数と修正後のプログラムのステップ数とが一致する必要があるが、上記第3実施例のマイクロコンピュータでは、プログラムの修正の前後でのステップ数を一致させないでも、プログラムの変更だけでなく、プログラムの追加や削除、別のプログラムの実行も可能になる。

なお、EROM 53は、プログラムメモリ2の全てのアドレスをカバーできるので、EROMの容量は少なくても済み、EROMを付加したことによる1チップマイクロコンピュータの価格の上昇は僅かで済む。

次に、EROM 53のデータ構成と割込み処理のフローについて、第6図および第7図を参照しながら説明する。EROM 53のプログラムデータ領域のデータ構成は、第6図に示すように、割込みを発生させるアドレスの入ったアドレスデータ

と、それにつながってコード部とアドレス・データ部とのペアが複数連続して書き込まれている。割込み処理では、第7図に示すフローのように、まず、コード部の値をチェックし、その値に応じて処理を選択する。コード部の値が「1」の時は、調整用のプログラムを実行し、コード部の値が「2」の時は、アドレス・データ部のアドレスのサブルーチンをコールし、コード部の値が「3」の時は、アドレス・データ部のアドレスにデータ部のデータを書き込み、コード部の値が「4」の時は、アドレス・データ部のアドレスのデータを読出し、コード部の値が「5」の時は、アドレス・データ部のアドレスにジャンプし、コード部の値が5以外の時は、リターンする。割り込み処理からのジャンプは、スタック（図示せず）に退避された戻り先のアドレスを書き換えてリターンすることで容易に行うことができる。勿論、EROM 53のデータを通常の命令コードに置き換えて実行させてもよいが、通常発生するバグは既に使われているサブルーチンを追加したり、数行の処理

をはぶくことで修正できることが多く、前記6つのコードがあれば修正できるので、割り込み処理のプログラムは非常に小さなフローで十分できる。もちろん、製品の必要に応じて割り込み処理のコードを追加してさらに柔軟性を持たせてもよい。

第8図は、前記調整用のプログラムの内容を示すフローチャートである。まず、調整フラグを「0」にセットする。この調整フラグは、調整器（図示せず）が接続されると「1」にセットされる。この調整器が接続されていないならばリターンし、接続されていると調整器との間で通信を行なう。この通信で使用する通信データは、第9図に示すように、調整コード部、アドレス部、データ部から構成されており、調整コード部の値が「1」の時は、アドレス・データ部のアドレスのサブルーチンをコールし、調整コード部の値が「2」の時は、アドレス・データ部のアドレスにデータ部のデータを書込み、調整コード部の値が「3」の時は、アドレス・データ部のアドレスのデータを調整器に送り、調整コード部の値が「4」

の時は、アドレス・データ部のアドレスにジャンプする。

調整コード部の値が「4」以外の時は、調整フラグが「1」かどうかをチェックし、調整フラグが「0」のままであればリターンし、調整フラグが「1」であれば通信を繰返す。調整器は、調整したい箇所のアドレスと調整コード「1」とをE R O M 53に書込み、調整用の通信が行われるのを待つ。製品は、E R O M 53のアドレスと実行しているプログラムのアドレスとが一致すると、割込みが発生し、調整コードが「1」なので通信を行なう。調整器は、調整したい内容を通信データに書込む。調整で複数のサブルーチンを実行させたい時は、初めの通信で調整コードを「1」に書換えて、複数回の通信でサブルーチンコールを繰返すことで行う。これにより、製品になった状態で製品のプログラムの全てのステップで調整が可能となる。そればかりでなく、調整箇所の追加もE R O M 53の内容を書換えることで簡単に行うことができる。

上記した第3実施例では、ソフトウェア割込みを行なう例を示したが、第10図に示す第4実施例ではハードウェア割込みを行なう例を示している。この第4実施例においては、アドレスデータ領域11i ($i = a, \dots, n$) およびベクターテーブル12i および一致検出手段13i の各1個を1組とする複数の組を持ち、アドレスデータ領域11i およびベクターテーブル12i はE R O Mに含まれており、アドレスデータ領域11i には割込みを発生したいアドレスが、ベクターテーブル12i には割込み処理の先頭アドレスが書込まれる。プログラムカウンタ1の値は、複数の組の一致検出手段13i に入力し、これらのうちどれかで一致が検出されると、一致したアドレスデータ領域11i と同じ組のベクターテーブル12i に出力許可信号を送り、そのベクターテーブル12i のデータ出力のみがプログラムカウンタ値変更手段14に伝えられる。

また、上記複数の組の一致検出手段13i の各出力はオア回路15を経て割込み発生回路16に

入力し、複数の組のうちどれかで一致が検出され、かつ、割込み許可フラグの内容が許可であると、割込みが発生し、プログラムカウンタ値変更手段14により、一致した組のベクターテーブル12i の値がプログラムカウンタ1に書込まれる。この時、同時に、プログラムカウンタ1の値がスタック（図示せず）に退避されることはいうまでもない。また、割込み許可フラグの内容が禁止であれば、割込みが発生せず、プログラムカウンタ1の値が変化しないことはいうまでもない。

第11図は、上記第4実施例におけるメモリマップを示しており、通常のプログラムはR O M領域に書込まれており、E R O Mは特に使用する必要はないが、プログラムの修正や調整などのためにR O M領域のプログラムを変更させたい場合は、その変更させたいアドレスをE R O M領域のアドレスデータ領域11a, 11b, 11c...に順次書込み、その時に行ないたい処理の先頭アドレスをベクターテーブル12a, 12b, 12c...に書込む。必要なサブルーチンがR O M領域に既に

ある場合には、そのサブルーチンの先頭アドレスを込めればよいが、必要なサブルーチンがROM領域にない場合には、EROM領域に新たにサブルーチンを追加し、その先頭アドレスをベクターテーブルに書き込む。

割込みを発生するアドレスの数はアドレスデータ領域11iの数で決まるが、通常の製品では5個もあれば十分である。

なお、前記各実施例において、書き込み制御部6、56の制御方式は、1チップマイクロコンピュータの外部端子(ポート端子など)を書込みモードにしたときに切換える兼用端子方式でもよく、命令デコード5に接続してプログラムで書き込む方式でもよい。

また、前記各実施例では、EROMはマイクロコンピュータに内蔵されていたが、EROMはマイクロコンピュータに外付け接続されていてもよい。また、前記各実施例とも、不揮発性メモリとしては、電源が常にバックアップされているシステムでは、通常のRAMも含むことはいうまでも

ない。

[発明の効果]

以上詳述したように本発明のマイクロコンピュータによれば、プログラムメモリの書き込みプログラムにバグなどが発見された場合やその一部を修正したい場合に、プログラムメモリの内容を書換えることなく、一部を疑似的に書換えてプログラムを修正することや、割込み処理を行わせることにより実質的にプログラムの追加とか削除を行ない、プログラムを修正することができる。

したがって、バグなどが発見されたマイクロコンピュータを再利用することができ、再度、プログラムメモリを作り直す必要がなくなり、生産工程に影響を与えないで済み、修正されたマイクロコンピュータを短期間に入手することができる。実際に、バグなどを修正する場合は、1~2行のプログラムを修正すれば済むことが多いので、本発明のマイクロコンピュータは極めて有用である。

4. 図面の簡単な説明

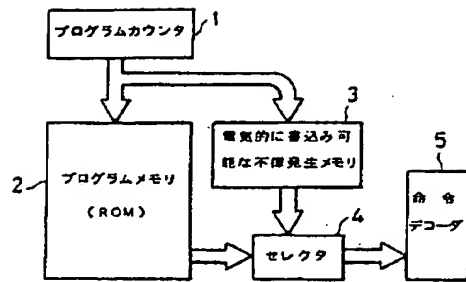
第1図は本発明のマイクロコンピュータの概念

を示す説明図、第2図は本発明のマイクロコンピュータの第1実施例を示すブロック図、第3図は第2図中のEROMの構成の一部の具体例を示す説明図、第4図は本発明のマイクロコンピュータの第2実施例に用いられるセレクトを示すブロック図、第5図は本発明のマイクロコンピュータの第3実施例を示すブロック図、第6図は第5図中のEROMのプログラムデータ領域のデータ構成を示す説明図、第7図は第5図のマイクロコンピュータにおける割込み処理の一例を示すフローチャート、第8図は第7図中の調整用のプログラムの内容を示すフローチャート、第9図は第8図中の調整用のプログラムの実行時における通信に使用されるデータの構成を説明する図、第10図は本発明のマイクロコンピュータの第4実施例を示すブロック図、第11図は第10図のマイクロコンピュータにおけるメモリマップを示す図、第12図は従来の1チップマイクロコンピュータを示すブロック図である。

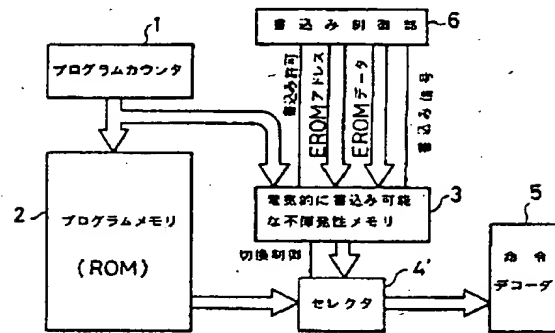
1…プログラムカウンタ、2…プログラムメモ

リ、3、3'、53…電氣的に書き込み可能な不揮発性メモリ、4、4'、4"…セレクト、5…命令デコード、6…書き込み制御部、11i…アドレスデータ領域、12i…ベクターテーブル、13i…一致検出手段、14…プログラムカウンタ値変更手段、15…オア回路、16…割込発生回路、32…アドレスデータ領域、33…プログラムデータ領域、34…データ判定回路、35…オア回路、41…データ判定回路、42…マルチプレクサ、54…一致検出部、55…割込み発生回路、56…書き込み制御部。

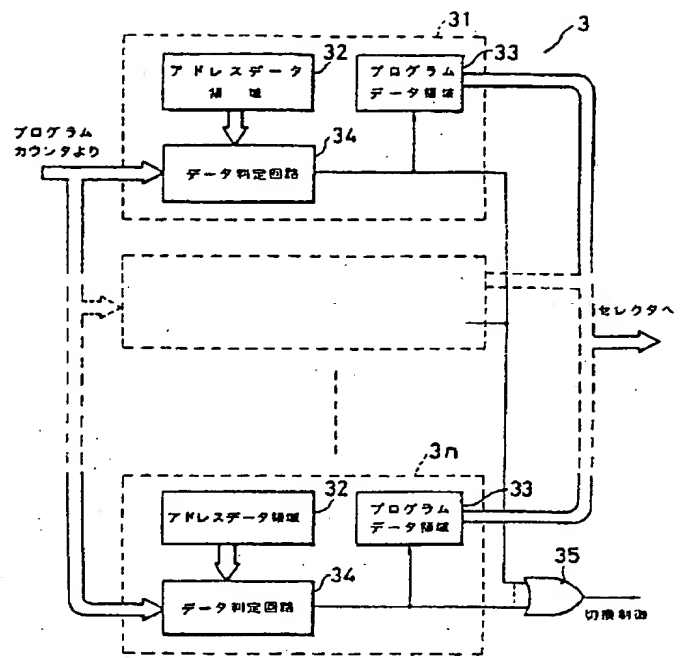
出願人代理人 弁理士 坪井 淳



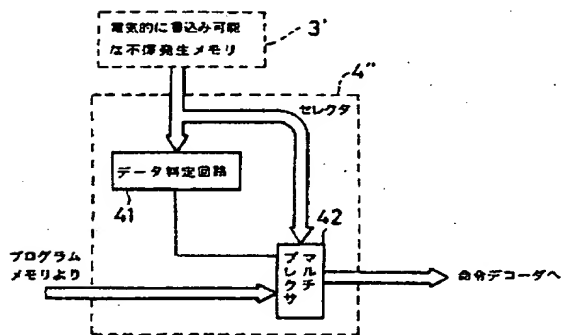
第 1 図



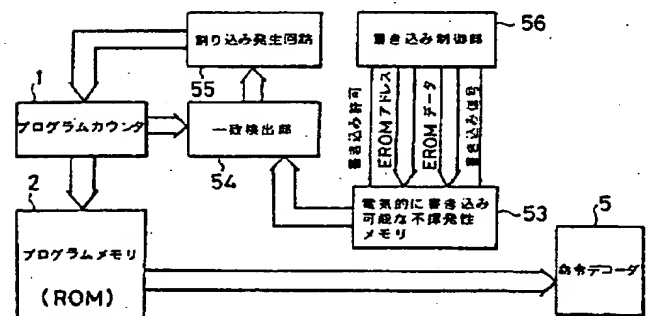
第 2 図



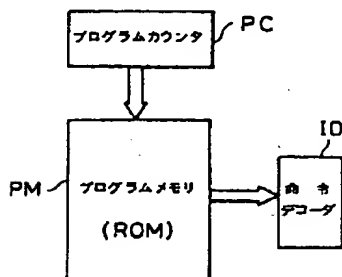
第 3 図



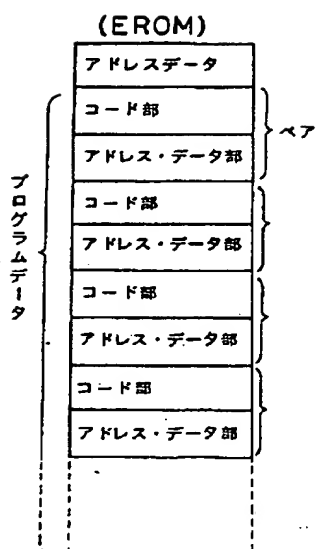
第 4 図



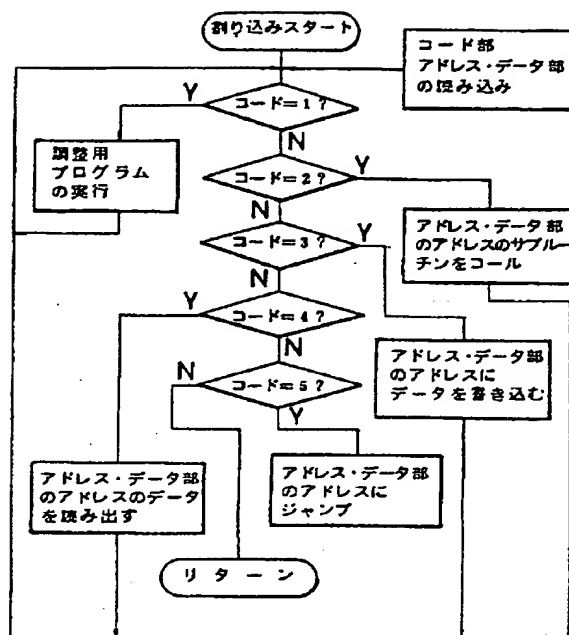
第 5 図



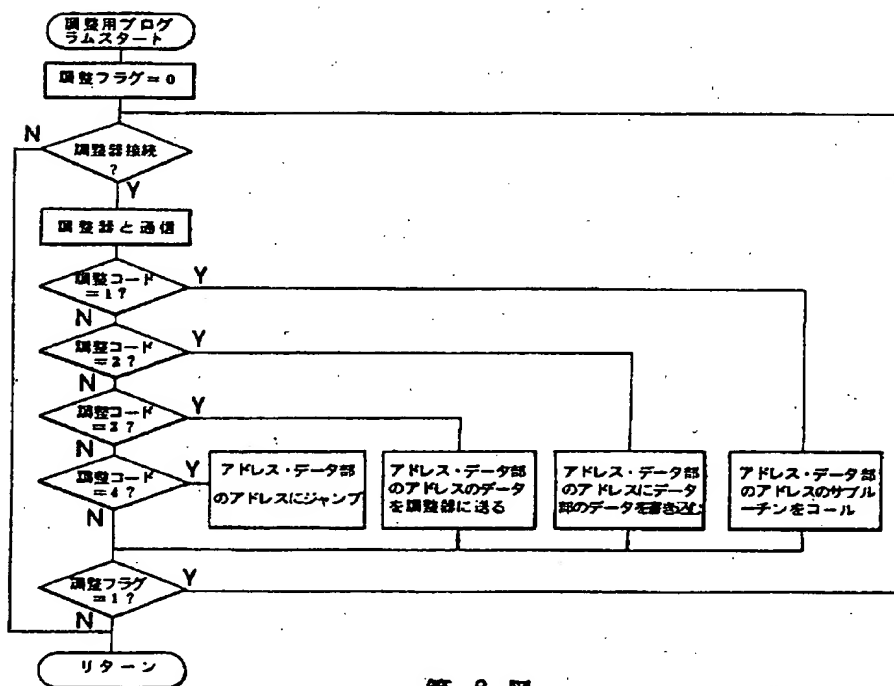
第 12 図



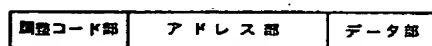
第 6 図



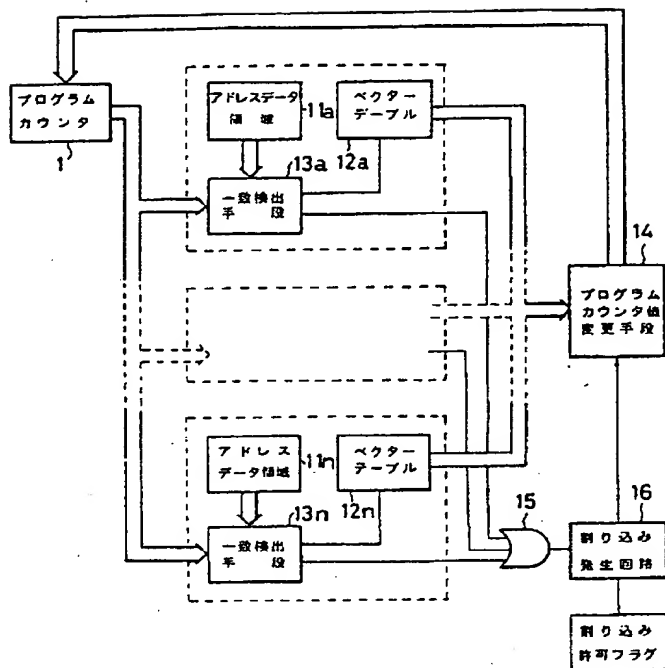
第 7 図



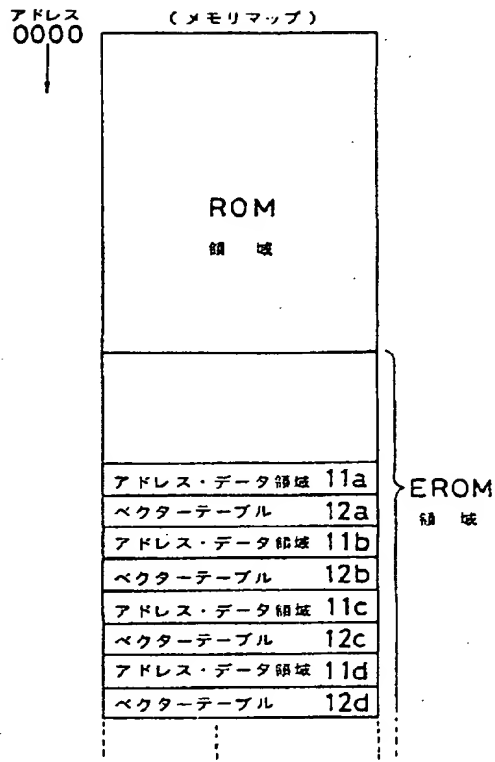
第 8 図



第 9 図



第 10 図



第 11 図

手続補正書

平成元年 10月12日

特許庁長官 吉田 文 毅 殿

1. 事件の表示

特願平 1-131861号

2. 発明の名称

マイクロコンピュータ

3. 補正をする者

事件との関係 特許出願人

(037) オリンパス光学工業株式会社

4. 代理人

東京都千代田区霞が関3丁目7番2号

〒100 電話 03(502)3181 (大代表)

(8881) 弁理士 坪 井 淳

5. 自発補正

6. 補正の対象

明細、図面

7. 補正の内容

(1) 明細書の第4頁第10行目に「アドレスデータデータ」とあるを「アドレスデータ」と訂正する。

(2) 明細書の第15頁第1行目に「EROM 32」とあるを「EROM 53」と訂正する。

(3) 明細書の第18頁第14行目に「5以外の時は、リターンする。」とあるを「5以外の時は、たとえば6のときにリターンする。」と訂正する。

(4) 明細書の第19頁第7行目に「調整フ1グ」とあるを「調整フラグ」と訂正する。

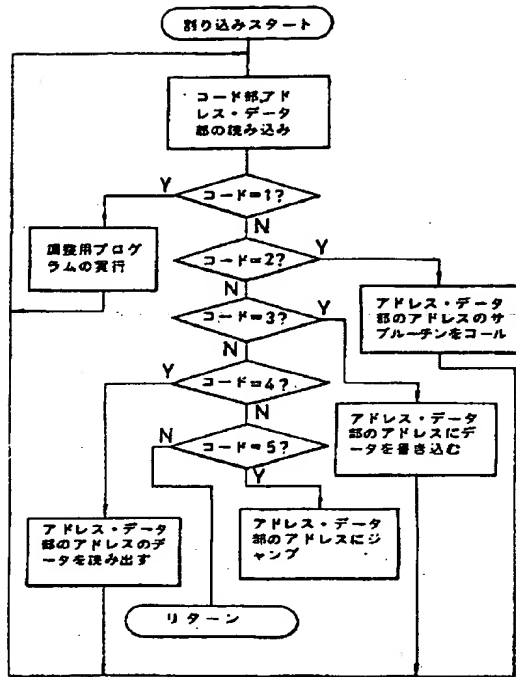
(5) 図面の第7図を別紙の通り訂正する。



手 続 補 正 書

平成 2 年 5.30 日

特許庁長官 吉 田 文 毅 殿



第 7 図

1. 事件の表示

特願平1-131861号

2. 発明の名称

マイクロコンピュータ

3. 補正をする者

事件との関係 特許出願人

(037) オリンパス光学工業株式会社

4. 代 理 人

東京都千代田区霞が関3丁目7番2号

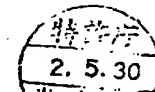
〒100 電話 03(502)3181 (大代表)

(6881) 弁理士 坪 井 淳

5. 自 発 補 正

6. 補正の対象

明細書



7. 補正の内容

(1) 明細書の第18頁第15行目に「ジャンプ」とあるを「リターン」と訂正する。

(2) 明細書の第19頁第8行目ないし第10行目にわたって「この調整フラグは、……セットされる。この調整器が」とあるを「次に、調整器（図示せず）が」と訂正する。

(3) 明細書の第20頁第6行目に「通信を繰返す。調整器は、」とあるを「通信を繰返す。なお、前述したように、始めに調整フラグは“0”にセットされているので、通常、調整処理は1つだけ行なわれる。連続的に複数回の調整処理を行なうときには、アドレスデータ部のアドレスにデータ部のデータを書込むステップにおいて、調整フラグの値を“1”に書込むように、調整器と通信のステップで調整器より通信を行ない、調整フラグの値を“1”にする。調整器は、」と訂正する。

(4) 同頁第9行目に「製品は、」とあるを「マイクロコンピュータは、」と訂正する。